SUMO User Conference 2023

Conference paper

https://doi.org/10.52825/scp.v4i.216

Published: 29 June 2023

Coping with Randomness in Highly Complex Systems Using the Example of Quantum-Inspired Traffic Flow Optimization

Maria Haberland[1] [https://orcid.org/0009-0009-4383-8632] and Lars Hohmuth[1][https://orcid.org/0009-0004-0360-7606]

[1] Fujitsu, Germany

Abstract. Developing new solutions to complicated large-scale problems typically requires large-scale numerical simulation. Therefore, traffic simulations often run against randomized simulations instead of real-world traffic situations. This paper demonstrates a method to calculate the statistical significance of numerical simulations and optimizations in the presence of numerous random variables in complex systems using one-sided paired t-tests. While the paper covers a specific Fujitsu traffic-optimization project which uses SUMO for simulating the traffic situation, the method can be applied to many similar projects where a complete investigation of the solution space is not feasible due to the size of the solution space.

Keywords: Statistics, Traffic simulation, Optimisation

1. Introduction

In 2020 the Hamburg Port Authority (HPA) initiated a traffic innovation project called MOZART. Its goal was to improve the car and heavy goods vehicle (HGV) traffic flow throughout the 30km road network of the port region by using an advanced digital twin, microscopic traffic simulations, and the Fujitsu Digital Annealer Unit (DAU) [1] to create a globally optimized signal plan for all 35 intersections once a minute. This signal plan will help the Port of Hamburg to achieve their part in the UN Sustainable Development Goals, specifically regarding the subgoals of “climate action by reducing pollution”, “responsible production and consumption by streamlining transport of goods”, “creating sustainable cities and communities by reducing traffic induces stressors”.

Fujitsu developed a solution concept that uses real time traffic simulation combined with multiple computationally generated alternative signal plans for each intersection. Using these the solution simulates the traffic between the intersections in parallel for all possible combinations of signal plans between adjacent intersections to calculate the coefficients of a stress function. This function is then written as a polynomial of quadratic order in binary variables called a Quadratic Unconstrained Binary Optimization (QUBO) suitable as input for the Digital Annealer. The solution then uses the Digital Annealer to find a global optimum of this stress function. For more details on the approach, see Traffic management through traffic signal control by Quantum-Inspired optimization. [2]

After validating the basic viability of the approach, this project was turned over to our team to develop a solution which can be run 24/7/365 in cities.

2. The TraFO System

The production application we are developing is called TraFO (Traffic Flow Optimization) and consists of a scalable ensemble of containers:

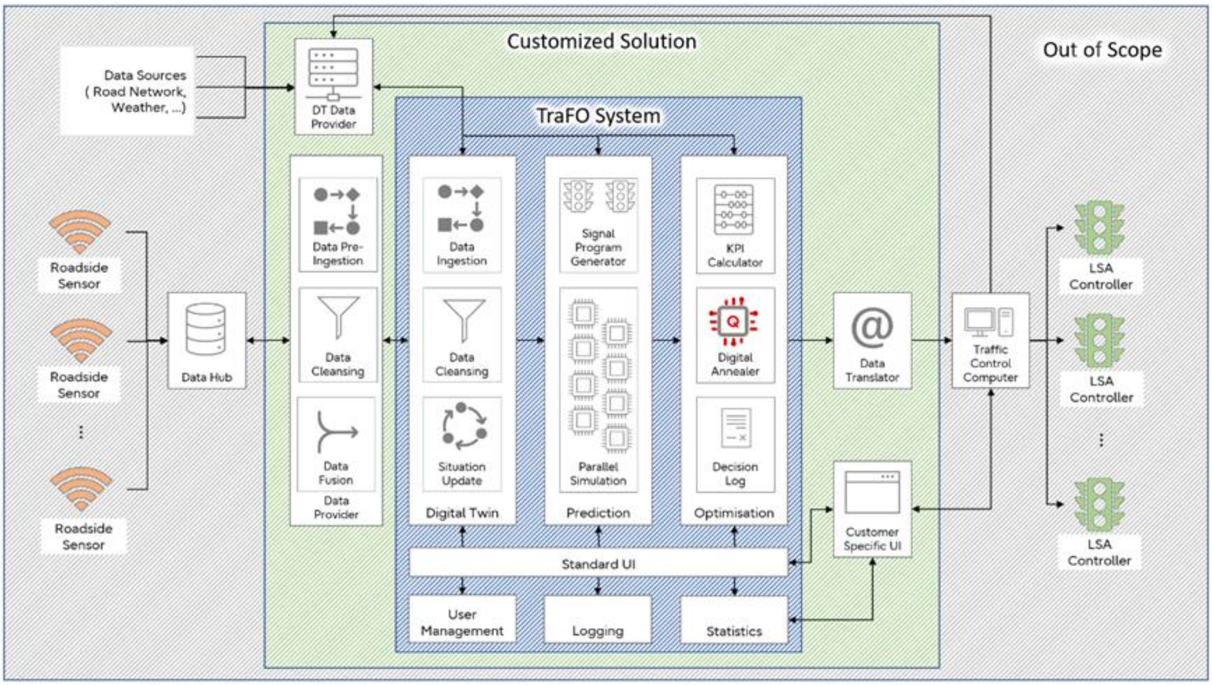


Figure 1. Structure of TraFO Optimization Platform

- The TraFO container sits at the center of the application. It performs the setup of the optimization, calculates the QUBO, calculates the KPIs, orchestrates the other containers, and provides the UI.
- The digital twin represents the real-world traffic. The digital twin is constructed using Vissim or SUMO networks and run in an instance of Vissim and SUMO [3] using traffic flow data for different time slots. We are using the available interface of these applications to collect the needed data for evaluation.
- Functions to do data ingestion and data cleaning for sensor data is also included into the digital twin to ensure a high data quality for the traffic simulations.
- Multiple instances can be used to run alternative scenarios in individually configured digital twins.
- The Signal Program Generator container calculates and preselects alternative traffic light programs to the TraFO for use by the short-term simulations. It also checks the compliance of the traffic light programs with legal and regulatory requirements. Currently, the Signal Program Generator implements the German regulatory requirements laid out in the Richtlinien für Lichtsignalanlagen (RiLSA), Edition 2015 [4]. This can be expanded by other regulatory requirements like MUTCD for the United States of America.

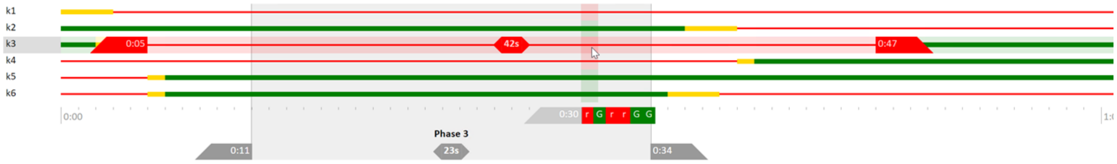
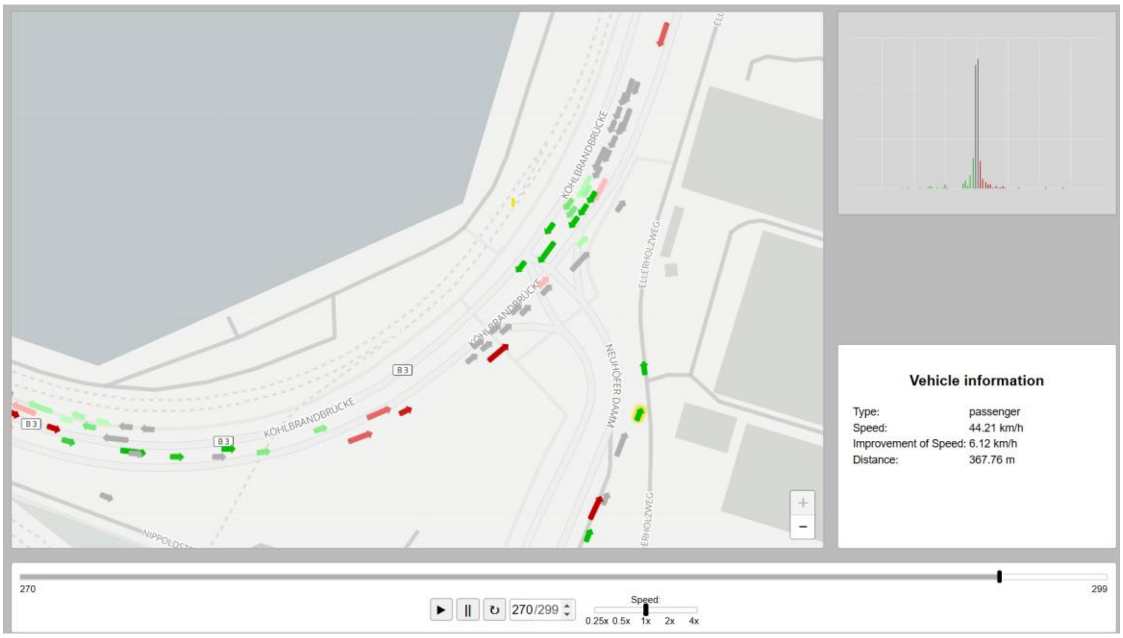


Figure 2. Visualization of Traffic Controller Programs

- To archive the required performance, the 125 short term simulations for each step are distributed across multiple instances, at least 10, running SUMO using libsumo in a containerized environment. The ideal number of instances depends on the available hardware, typically one per available core.
- The Digital Annealer calculates the optimization result.
- A standard UI is provided to consolidate the results and configure the different components. Customer specific UI, for example to display customer specific KPIs or add external data, can be added



Vehicle information

Type: passenger
Speed: 44.21 km/h
Improvement of Speed: 6.12 km/h
Distance: 367.76 m

Figure 3. UI for Digital Twin Visualisation

- We are using a containerized MongoDB to store the road network and other input data as well as the simulation results, vehicle trajectories and KPIs.

In each optimization cycle, the TraFO container:

1. Collects the up-to-date traffic situation and signal program from the Digital Twin.
2. Passes them to the short-term simulations together with the possible signal programs from the signal program generator and starts the short-term simulations.
3. Builds the QUBO with the results of the short-term simulations.
4. Passes the QUBO to the DAU.
5. Collects the optimal signal programs for each signal head from the DAU.
6. And passes them to the long-term simulation at the start of the next cycle.

2.1 Randomness in TraFO

Since we cannot develop an application in live traffic, we must rely on traffic simulation tools to create a digital twin. In our project we use two different simulation tools: SUMO and PTV Vissim. Both tools use random number generators (RNG) to place vehicles and simulate driver behavior in traffic. Both also let users specify random seeds to reproduce simulation runs.

Additionally, in TraFO, the generation of signal program alternatives and the short-term simulation use multiple random number generator instances to decouple different simulation aspects, including

- randomness when loading vehicles (vehicle type distributions, speed deviations, ...)
- probabilistic flows
- vehicle driving dynamics

and more.

Changes to the random seeds can lead to widely varying traffic flows as shown below. Depending on the traffic flow, one and the same alternate signal plan can be highly effective, or highly detrimental to the flow of traffic.

In our project, we came across the following types of random number generators:

1. Pseudo Random Number Generators in SUMO
   Sumo uses the Mersenne Twister [5] is a general-purpose pseudorandom number generator developed in 1997 by Makoto Matsumoto (松本 眞) and Takuji Nishimura (西村拓士). This algorithm is widely used by commercial software like Microsoft Excel, SAS, SPSS, and Matlab as well as standard libraries like the standard C++ library, CUDA, and the NAG Numerical Library.
2. Pseudo Random Number Generators in Vissim
   PTV Vissim is a proprietary commercial software, so while it uses multiple random number generators to vary the patterns of stochastic assignments and traffic signals, the exact type could not be determined from the technical documentation.

3. How Big is the Influence of Random Numbers?

The number of possible combinations of random seeds between the long-term simulation and optimization is so large (~ $3.4*10^{38}$), that exhaustive sampling is impossible (~10^{28} years at 1000 simulations per second). Additionally, we are using about 10 parameters to tune the short-term simulation, which further inflates the search space.

3.1 Measuring Traffic Quality

To score the simulation runs we chose four key performance indications (KPI) for comparing the different runs:

- Average Speed: at every simulation step, the average speed of all vehicles in the network is calculated and then averaged again over the run time of the simulation. A higher value is considered better.
- Number-Of-Vehicles: at every simulation step the number of vehicles in the network is counted. The KPI is the average of all these sums. A lower value is considered better, because when having the same number of vehicles getting into the network that means more vehicles already leaving the network earlier.
- Deceleration: at every simulation step, the average deceleration of all vehicles in the network is calculated. Deceleration in this case means, that only negative acceleration values were considered, using 0 for positive accelerations. This is used, to get value more fluent traffic, as stop and go creates much more fuel consumption and CO_2 production than a slower but steady traffic. At the end the values were averaged again over the run time of the simulation. A lower value is considered better.
- Traffic-Jam: at every simulation step, the vehicles which drive at less than 5 km/h are counted. The Traffic-Jam KPI is the average of all these sums. A lower value is considered better.

All KPIs are only calculated after a 900-second start-up phase without optimizations which is needed to populate each part of the network with enough vehicles to generate valid data.

To understand the impact the randomness in simulating traffic, we started with an analysis of baseline simulations of the same situation using different random seeds.

Further KPIs can be added to the list as long as they are numerical and metric.

3.2 Baseline Comparisons

To get an impression of the difference between simulations of the same initial situation using different random seeds, we have selected two simulations in our application, showing their KPIs and the values for average speed over simulation time: once initialized with 9299 as the random seed and once with 9340.

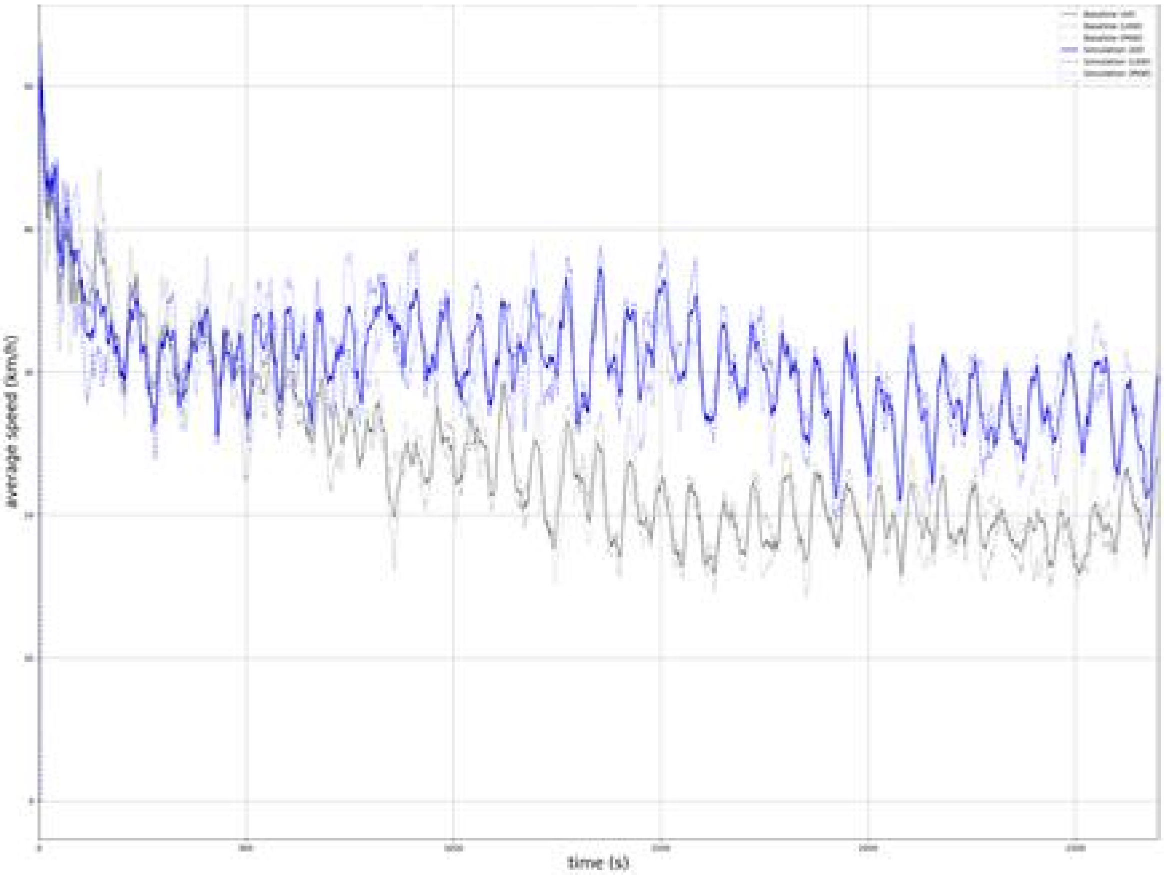


Figure 4. Comparison of average speed over simulation time of two simulations

The average speed of these two simulations has a ratio of almost 2:3 just because of the randomness of the input data for the simulation. Given that an average improvement of about 10% would be considered a huge success in the real world, any optimization effect would be buried by the differences created by the choice of the random seed.

Table 1. KPI of baseline 9299 and 9340.

Simulation Seed	Average Speed	Number Of Vehicles	Deceleration	Traffic Jam
9299	29.36	203.2	-0.791	65.1
9340	20.51	289.7	-0.680	143.5

To get a better impression, we created 50 baseline simulations and generated a boxplot (see Fig. 5) from the generated data.

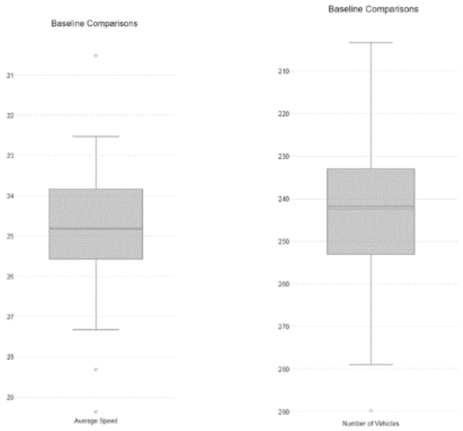


Figure 5. Boxplot for baseline simulations

As one can see, the measured KPIs vary widely even though we haven't used the digital annealer and haven't generated any optimization or changed any traffic light program. The big question is now: If we already have such big differences in the baseline simulations, how could we prove that our optimization creates better results? Just running some optimizations for some randomly picked random seeds wouldn't prove anything, even though one might get a decent hunch from running a few thousand simulations.

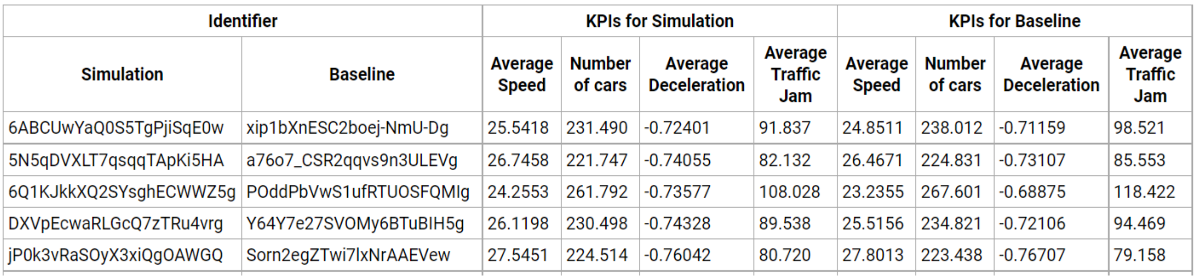
4. Are We Really Improving Traffic?

It would already be a huge success if we could improve the traffic flow by about 10%. As shown above, two baseline simulation can already have a bigger difference using the same pool of traffic light programs, caused just by variations in simulating the behavior of drivers. As each round of optimization creates a change in the overall simulated system, the vehicle behavior will be different over time. For example, if without optimization a car would have been waiting at a crossing, it is possible that the optimization has it now already driving past the intersection. This will influence the later simulation of other vehicles as well.

Therefor it is not sufficient to just pick a few simulations and compare the baseline with the optimized one, but instead use a more sophisticated approach.

4.1 Medical Science to the Rescue

Our solution for this dilemma was to look at other discipline in sciences. How do they handle such uncertainties? After reviewing some approaches, we selected methods used in medical sciences and pharmaceutical testing [6]. These seem suitable, because the traffic simulation



Identifier		KPIs for Simulation				KPIs for Baseline			
Simulation	Baseline	Average Speed	Number of cars	Average Deceleration	Average Traffic Jam	Average Speed	Number of cars	Average Deceleration	Average Traffic Jam
6ABCUwYaQ0S5TgPjiSqE0w	xip1bXnESC2boej-NmU-Dg	25.5418	231.490	-0.72401	91.837	24.8511	238.012	-0.71159	98.521
5N5qDVXLT7qsqqTApKi5HA	a76o7_CSR2qqvs9n3ULEVg	26.7458	221.747	-0.74055	82.132	26.4671	224.831	-0.73107	85.553
6Q1KJkkXQ2SYsghECWWZ5g	POddPbVwS1ufRTUOSFQMlg	24.2553	261.792	-0.73577	108.028	23.2355	267.601	-0.68875	118.422
DXVpEcwaRLGcQ7zTRu4vrg	Y64Y7e27SVOMy6BTuBIH5g	26.1198	230.498	-0.74328	89.538	25.5156	234.821	-0.72106	94.469
jP0k3vRaSOyX3xiQgOAWGQ	Sorn2egZTwi7lxNrAAEVew	27.5451	224.514	-0.76042	80.720	27.8013	223.438	-0.76707	79.158

Figure 6. Raw statistical data generated by simulation

programs essentially model human behavior and must deal with large solution spaces and incomplete knowledge of the “participants” in large scale medical studies as well.

When doing new drug development or studies for new therapies, biometric statistical methods are used. Similar statistics are also used in human studies in psychology, for example when evaluating therapies or clustering human behavior. So, we took a detailed look at how they solve the problem of distributed data in randomized testing.

Any study normally consists of three steps: the design of the study, the execution, and the evaluation. We will use the same three steps for our problem.

4.2 Design of the Study

In the design phase, one answers the following questions:

- Which hypothesis(es) do we want to prove?
- Which type of study do we conduct?
- What aspects will we measure during execution?
- How will we evaluate the generated data?
- How many “participants” do we need?

The hypothesis we want to prove is that using the optimization with the QUBO and the digital annealer results in better traffic flow. In other words, we want to check whether the traffic in the optimized simulation is more fluid than the baseline simulation. In statistics, one uses an inverted null hypothesis, which is: The traffic flow after optimization is not more fluid than the traffic flow in the baseline simulation.

For the other prescribed steps, we build an analogy to medical studies. When we look at our generated data, we see that we can run a baseline simulation and an optimization simulation using the same random seeds. This is reminiscent of “twin studies” in medical sciences. So, we ran baseline simulations and optimization simulations using the same random seed for our testbed and a fixed set of parameters for our optimization and calculated the above mentioned KPIs.

To test whether one set of results were better than the other, we chose a one-sided paired t-test [6], a popular method to compare data of twin studies in medical biometry.

One initial step is to calculate how many “participants” are needed for a statistically significant result. Using standard online tools, a Cohens d of 0.2 (the minimal feasible value) required 156 “participants”[7], [8]. This tells us that we had to run at least 312 (156 * 2) simulations to achieve a useful result. Due to the parallel execution of the simulations in TraFO, this could be done in an acceptable length of time.

4.3 Execution of the Study

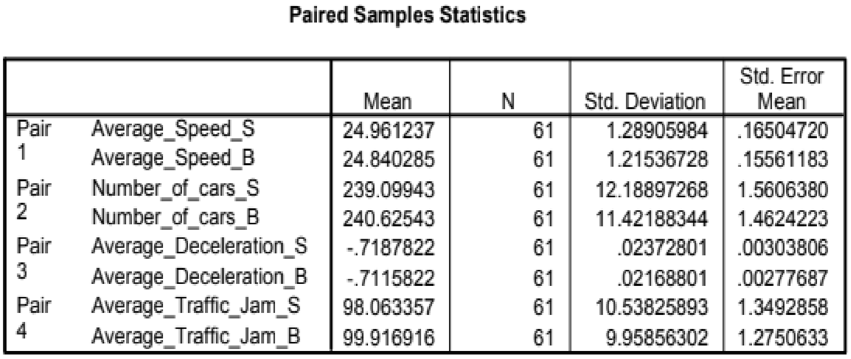
For the execution we had to run a lot of simulations and collect the data (see Figure 6). We had previously implemented a batch mode TraFO, which allows us to run a defined number of pairs of simulations (baseline and optimization) for a specific situation using different randomly generated random seeds. All data is saved in a MongoDB database and can be processed later.

Because running simulations in our optimized version of SUMO is much faster than in Vissim, we focused our first evaluation runs on SUMO simulations. Using SUMO, we can run a single simulation of a 45-minute traffic flow in about 5 minutes with 11 parallel simulation threads on a Fujitsu CELSIUS J5010 (Intel Core i7-10700, 32GB RAM) so we would have needed about one day to run all the simulations we needed.

To save time, we separated the simulation runs into subsets, so that we could start evaluating data even while additional simulations were still running. For our first run we chose a set of optimization parameters which caused some improvements in traffic flow in randomly selected trial runs.

4.4 Statistical Evaluation of the Study

For the first evaluation we use SPSS to calculate the statistical results.

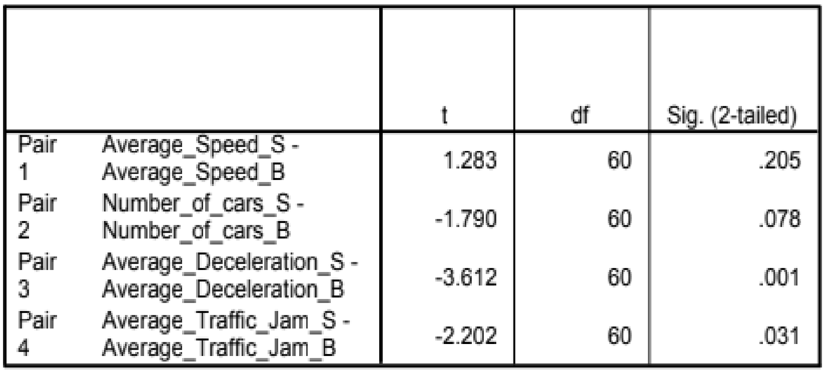


Paired Samples Statistics

		Mean	N	Std. Deviation	Std. Error Mean
Pair 1	Average_Speed_S	24.961237	61	1.28905984	.16504720
	Average_Speed_B	24.840285	61	1.21536728	.15561183
Pair 2	Number_of_cars_S	239.09943	61	12.18897268	1.5606380
	Number_of_cars_B	240.62543	61	11.42188344	1.4624223
Pair 3	Average_Deceleration_S	-.7187822	61	.02372801	.00303806
	Average_Deceleration_B	-.7115822	61	.02168801	.00277687
Pair 4	Average_Traffic_Jam_S	98.063357	61	10.53825893	1.3492858
	Average_Traffic_Jam_B	99.916916	61	9.95856302	1.2750633

Figure 7: Mean and standard derivation for each data set

When running our first evaluation using SPSS, we were surprised to see that all averages were slightly better in the optimization simulations than in the baseline simulations already (see Figure 7). Furthermore, we already got statistically significant metrics (Sig < 0.05) for the Traffic-Jam KPI and a statistical trend (Sig < 0.1) for the Number-of-Cars KPI (see Figure 8). The results can be reproduced using the Excel implementation for the t-Test [9].



		t	df	Sig. (2-tailed)
Pair 1	Average_Speed_S - Average_Speed_B	1.283	60	.205
Pair 2	Number_of_cars_S - Number_of_cars_B	-1.790	60	.078
Pair 3	Average_Deceleration_S - Average_Deceleration_B	-3.612	60	.001
Pair 4	Average_Traffic_Jam_S - Average_Traffic_Jam_B	-2.202	60	.031

Figure 8. t-test results

This shows that even with the first selected optimization configuration and our simple KPIs, we can show that our algorithm improves the traffic flow. Calculating Cohens d we can even show that the results are not only significant but also show a statistical effect. We expect that we will find other optimizations with even larger effects in the future.

Because we want to run these steps with more optimization configurations in future, we added an implementation of the one-sided pair-test to TraFO using the SciPy library for Python. This will allow us to run more tests faster and displaying the results inside TraFO.

5. Conclusion and Outlook

Because of the strong dependency of complex systems on pseudo random number generator seeds and initial conditions, it is often hard to separate effects of deliberate optimizations in digital simulations from spurious changes due to varying initial conditions. This becomes a real issue if the simulation of the real-world system relies on multiple independent simulations using random number generators. Traffic flow simulations often shows this behavior.

This paper shows that using biometric statistical evaluation of a large number of simulations for a set of simulation parameters can help prove that a specific computational approach is working and can also eliminate unsuitable combinations of parameters. Compared to the common approach to verify the effectiveness of new computational approaches in traffic simulation by “running a few experiments” or “eyeballing it”, it increases the confidence in the accuracy of the simulations and optimizations and makes it easier to explain the approach and its benefits to future users with hard statistical evidence.

Additionally, these measures - for example Cohens d - will help identify the best set of optimization parameters early in research and simulation projects, so researchers and traffic engineers can home in on strategies that provide the highest added value.

We will use one-sided paired t-tests in more simulations for our scenario with different configurations to evaluate the effects of different tweaks to the optimization algorithm. Our experiments already showed that it is possible to get a statistically significant improvement in our scenario at the port of Hamburg using the digital annealer. We are already using this approach to find the best parameter sets for Hamburg as well as for other situations.

For example, we have now a tool kit to check quickly, whether more complex scoring algorithms or new optimization algorithm leads to better solutions. We already have a set of different methods for generating QUBOs for the digital annealer which we now can compare to each other quantitatively.

In the future we want to use this approach in other projects, which also simulate scenarios based on human behavior or pseudo random number generators, including large scale SUMO simulations. We also want to encourage other SUMO users to use it as well.

We also will continue to explore additional statistical tests for more detailed analysis with the Wilcoxon signed-rank test as the next candidate.

Data availability statement

Due to the nature of the research, due to commercial supporting data is not available.

Author contributions

Maria Haberland: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Visualization, Writing. **Lars Hohmuth**: Conceptualization, Project administration, Validation, Visualization, Writing.

Competing interests

The authors declare that they have no competing interests.

Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

Acknowledgement

We would like to thank the whole development team for the TraFO solution for building up this wonderful platform, that allows us to create this work. Especially we would like to thank Peter Marschke for his work providing services to use the SUMO interfaces and his support using them. Furthermore, we would like to thank Frank Hambach for his work on the traffic light program generation, Hannah Berlau for her help creating beautiful UIs and Mike Maaßen for his development of further functions.

References

1. Aramon, M., Rosenberg, G., Valiante, E., Miyazawa, T., Tamura, H., & Katzgraber, H., Physics-Inspired Optimization for Quadratic Unconstrained Problems Using a Digital Annealer. Front. Phys., 05 April 2019, doi: https://doi.org/10.3389/fphy.2019.00048.
2. F. Schinkel, I. Schwende, R. Schade, E. Cerny, M. Fellendorf, Traffic management through traffic signal control by Quantum-Inspired optimization. 27th ITS World Congress, Hamburg, Germany, 2021.
3. Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, Evamarie Wießner, Microscopic Traffic Simulation using SUMO. IEEE Intelligent Transportation Systems Conference (ITSC), 2018, doi: https://doi.org/10.1109/ITSC.2018.8569938.
4. Forschungsgesellschaft für Straßen- und Verkehrswesen. Richtlinien für Lichtsignalanlagen - Lichtzeichenanlagen für den Straßenverkehr. Berlin: FGSV. 2015.
5. M. Matsumoto, T. Nishimura, Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. ACM Transactions on Modeling and Computer Simulation, 1998, doi: https://doi.org/10.1145/272991.272995.
6. J. Hedderich, L. Sachs, Hypothesentest, Angewandte Statistik. Berlin, Heidelberg: Springer Spektrum, 2018.
7. J. Frost, Cohens D: Definition, Using & Examples, Statistics by Jim: https://statisticsbyjim.com/basics/cohens-d/, (04/2023).
8. Hemmerich, W., Statistik Guru: Cohen’s d für den gepaarten t-Test berechnen, Statistics Guru: https://statistikguru.de/rechner/cohens-d-gepaarter-t-test.html, (04/2023).
9. B. Walther, T-Test bei abhängigen Stichproben in Excel durchführen, https://bjoern-walther.com/t-test-bei-abhaengigen-stichproben-in-excel/, (04/2023).